medium 15); system 31 (ACE 100 of wireless device 32 with configuration information or modules 70 stored in a form of memory (separately illustrated in Figure 3), such as RAM or a matrix interconnection network ("MIN"), discussed below); system 21 (ACE 100 of computer 55B with configuration information or modules 70 stored in computer readable medium 20); system 22 (ACE 100 of server 54B with configuration information or modules 70 stored in a form of memory (separately illustrated in Figure 3)); and system 23 (ACE 100 of router 53B with configuration information or modules 70 stored in a memory (separately illustrated in Figure 3)). As may be apparent, a system of the present invention may be embodied within any device or other article, in addition to those illustrated (*e.g.*, LAN 41, wireless LAN 43, WAN 42, and adjunct network entity 50), which include both an ACE 100 and configuration information (or module 70) for the provision of a corresponding operating mode, and may otherwise be co-extensive with any particular apparatus or other embodiment.

On Page 14, starting on line 1, please replace the paragraph with the following:

The interface 62 is utilized for appropriate connection to a relevant channel, network or bus; for example, the interface 62 may provide impedance matching, drivers and other functions for a wireline interface, may provide demodulation and analog to digital conversion for a wireless interface, and may provide a physical interface for the memory 61 with other devices. In general, the interface 62 is used to receive and transmit data, depending upon the selected embodiment, such as voice information, configuration information, silverware modules (70), control messages, authentication data and other pertinent information. The ACE 100 may also be configured to provide the functionality of the interface 62, including internal IC input/output ("I/O") and external (off-chip) I/O, such as for PCI bus control. The memory 61 may be an integrated circuit or portion of an integrated circuit, such as various forms of RAM, DRAM, SRAM, FeRAM, MRAM, ROM, EPROM, E^2PROM, flash, and so on. For non-IC (or non-SOC) embodiments, the memory 61 may also be a magnetic (hard of floppy) drive, an optical storage device, or any other type of data storage apparatus and, as indicated above, may be distributed across multiple devices. In addition, depending upon the selected embodiment, and as discussed in greater detail below, the memory 61 may

also be included within the ACE 100, through memory computational elements or within the matrix interconnection network (MIN). One or more processing elements 65 optionally may be included within system 60, to provide any additional processing capability, such as reduced instruction set ("RISC") processing, or may be included as computational elements within the ACE 100.

On Page 14, starting on line 24, please replace the paragraph with the following:

Figure 4 is a block diagram illustrating a preferred ACE apparatus 100 embodiment in accordance with the present invention. The ACE 100 is preferably embodied as an integrated circuit, or as a portion of an integrated circuit having other, additional components. (The ACE 100 is also described in detail in the related application.) In the preferred embodiment, and as discussed in greater detail below, the ACE 100 includes one or more reconfigurable matrices (or nodes) 150, such as matrices 150A through 150N as illustrated, and a matrix interconnection network (MIN) 110. Also in the preferred embodiment, and as discussed in detail below, one or more of the matrices 150, such as matrices 150A and 150B, are configured for functionality as a controller 120, while other matrices, such as matrices 150C and 150D, are configured for functionality as a memory 140. While illustrated as separate matrices 150A through 150D, it should be noted that these control and memory functionalities may be, and preferably are, distributed across a plurality of matrices 150 having additional functions to, for example, avoid any processing or memory "bottlenecks" or other limitations. Such distributed functionality, for example, is illustrated in Figure 5. The various matrices 150 and matrix interconnection network 110 may also be implemented together as fractal subunits, which may be scaled from a few nodes to thousands of nodes. As mentioned above, in the preferred embodiment, the adjunct network entity 50 of the present invention is embodied as an ACE 100 or as one or more matrices 150 (with corresponding interconnection networks).

On Page 15, starting on line 13, please replace the paragraph with the following:

A significant departure from the prior art, the ACE 100 does not utilize traditional (and typically separate) data, direct memory access ("DMA"), random access,

configuration and instruction busses for signaling and other transmission between and among the reconfigurable matrices 150, the controller 120, and the memory 140, or for other I/O functionality. Rather, data, control (such as power and timing information) and configuration information are transmitted between and among these matrix 150 elements, utilizing the matrix interconnection network 110, which may be configured and reconfigured, to provide any given connection between and among the reconfigurable matrices 150, including those matrices 150 configured as the controller 120 and the memory 140, as discussed in greater detail below.

On Page 16, starting on line 5, please replace the paragraph with the following:

The matrices 150 configured to function as memory 140 may be implemented in any desired or preferred way, utilizing computational elements (discussed below) of fixed memory elements, and may be included within the ACE 100 or incorporated within another IC or portion of an IC (such as memory 61). In the preferred embodiment, the memory 140 is included within the ACE 100, and preferably is comprised of computational elements which are low power consumption random access memory (RAM), but also may be comprised of computational elements of any other form of memory, such as flash, DRAM, SRAM, MRAM, ROM, EPROM or E^2PROM. As mentioned, this memory functionality may also be distributed across multiple matrices 150, and may be temporally embedded, at any given time, as a particular MIN 110 configuration. In addition, in the preferred embodiment, the memory 140 preferably includes DMA engines, not separately illustrated.

On Page 17, starting on line 3, please replace the paragraph with the following:

The matrix interconnection network 110 of Figure 4, and its subset interconnection networks separately illustrated in Figures 5 and 6 (Boolean interconnection network 210, data interconnection network 240, and interconnect 220), collectively and generally referred to herein as "interconnect", "interconnection(s)", "interconnection network(s)" or MIN, may be implemented generally as known in the art, such as utilizing field programmable gate array ("FPGA") interconnection networks or switching fabrics, albeit in a considerably more varied fashion. As used herein, "field

programmability" refers to the capability for post-fabrication adding or changing of actual IC functionality, as opposed to programming of existing IC structure or function (such as in a microprocessor or DSP). In the preferred embodiment, the various interconnection networks are implemented as described, for example, in U.S. Patent No. 5,218,240, U.S. Patent No. 5,336,950, U.S. Patent No. 5,245,227, and U.S. Patent No. 5,144,166, and also as discussed below and as illustrated with reference to Figures 8, 9 and 10. These various interconnection networks provide selectable (routable or switchable) connections between and among the controller 120, the memory 140, the various matrices 150, and the computational units 200 and computational elements 250 discussed below, providing the physical basis for the configuration and reconfiguration referred to herein, in response to and under the control of configuration signaling generally referred to herein as "configuration information" (and provided in modules 70). In addition, the various interconnection networks (110, 210, 240 and 220) provide selectable or switchable data, input, output, control and configuration paths, between and among the controller 120, the memory 140, the various matrices 150, and the computational units 200 and computational elements 250, in lieu of any form of traditional or separate input/output busses, data busses, DMA, RAM, configuration and instruction busses.

On Page 21, starting on line 13, please replace the paragraph with the following:

Next, the present invention also utilizes a tight coupling (or interdigitation) of data and configuration (or other control) information, within a plurality of packets or within one, effectively continuous stream of information. This coupling or commingling of data and configuration information, referred to as "silverware" or as a "silverware" module, is illustrated in Figure 1. This coupling of data and configuration information into one information (or bit) stream, which may be continuous or divided into packets, helps to enable real-time reconfigurability of the ACE 100, without a need for the (often unused) multiple, overlaying networks of hardware interconnections of the prior art. For example, as an analogy, a particular, first configuration of computational elements 250 at a particular, first period of time, as the hardware to execute a corresponding algorithm during or after that first period of time, may be viewed or conceptualized as a hardware

analog of "calling" a subroutine in software which may perform the same algorithm. As a consequence, once the configuration of the computational elements 250 has occurred (*i.e.,* is in place), as directed by (a first subset of) the configuration information, the data for use in the algorithm is immediately available as part of the silverware module. Referring to Figure 1, this is illustrated by "calling" various configurations (through references or flags in fields 80 and 81, for example, for configurations "A" and "B"), closely followed by providing the data for use in these configurations (fields 83 and 84 for configuration "A", fields 86 and 87 for configuration "B"). The same computational elements 250 may then be reconfigured for a second period of time, as directed by second configuration information (*i.e.,* a second subset of configuration information), for execution of a second, different algorithm, also utilizing immediately available data. The immediacy of the data, for use in the configured computational elements 250, provides a one or two clock cycle hardware analog to the multiple and separate software steps of determining a memory address and fetching stored data from the addressed registers. This has the further result of additional efficiency, as the configured computational elements 250 may execute, in comparatively few clock cycles, an algorithm which may require orders of magnitude more clock cycles for execution if called as a subroutine in a conventional microprocessor or digital signal processor ("DSP").

Page 22, starting on line 25, please replace the paragraph with the following:

Referring again to Figure 4, the functions of the controller 120 (preferably matrix (KARC) 150A and matrix (MARC) 150B, configured as finite state machines) may be explained (1) with reference to a silverware module, namely, the tight coupling of data and configuration information within a single stream of information, (2) with reference to multiple potential modes of operation, (3) with reference to the reconfigurable matrices 150, and (4) with reference to the reconfigurable computation units 200 and the computational elements 250 illustrated in Figure 5. As indicated above, through a silverware module, the ACE 100 may be configured or reconfigured to perform a new or additional function, such as an upgrade to a new technology standard or the addition of an entirely new function, such as the addition of a music function to a mobile communication device. Such a silverware module may be stored in the matrices 150 of

memory 140, or may be input from an external (wired or wireless) source through, for example, matrix interconnection network 110. In the preferred embodiment, one of the plurality of matrices 150 is configured to decrypt such a module and verify its validity, for security purposes. Next, prior to any configuration or reconfiguration of existing ACE 100 resources, the controller 120, through the matrix (KARC) 150A, checks and verifies that the configuration or reconfiguration may occur without adversely affecting any pre-existing functionality, such as whether the addition of music functionality would adversely affect pre-existing mobile communications functionality. In the preferred embodiment, the system requirements for such configuration or reconfiguration are included within the silverware module or configuration information, for use by the matrix (KARC) 150A in performing this evaluative function. If the configuration or reconfiguration may occur without such adverse affects, the silverware module is allowed to load into the matrices 150 (of memory 140), with the matrix (KARC) 150A setting up the DMA engines within the matrices 150C and 150D of the memory 140 (or other stand-alone DMA engines of a conventional memory). If the configuration or reconfiguration would or may have such adverse affects, the matrix (KARC) 150A does not allow the new module to be incorporated within the ACE 100.

On Page 28, starting on line 6, please replace the paragraph with the following:

In the preferred embodiment, the selection of various input and output lines 281 and 291, and the creation of various connections through the interconnect (210, 220 and 240), is under control of control bits 265 from a computational unit controller 255, as discussed below. Based upon these control bits 265, any of the various input enables 251, input selects 252, output selects 253, MUX selects 254, DEMUX enables 256, DEMUX selects 257, and DEMUX output selects 258, may be activated or deactivated.

On Page 28, starting on line 13, please replace the paragraph with the following:

The exemplary computation unit 200 includes the computation unit controller 255 which provides control, through control bits 265, over what each computational element 250, interconnect (210, 220 and 240), and other elements (above) does with every clock cycle. Not separately illustrated, through the interconnect (210, 220 and 240), the various